

Week 7 - Assignment

Programming for Data Science 2024

Exercises for the topics covered in the seventh lecture.

The exercise will be marked as passed if you show a reasonable understanding of the plotting concepts covered in the lecture. You do **not** have to make a perfect replica of the example plots to pass.

The exercises must be handed in via **ILIAS** (Homework assignments). Deliver your submission as a compressed file (zip) containing:

- one .py or .ipynb file with code
- the created figure as a *png* file.

The name of the .zip **and** the .py/.ipynb file **and** the png file **must** be *SurnameName* of the two members of the group. Example: Riccardo Cusinato + Athina Tzovara = *CusinatoRiccardo_TzovaraAthina.zip* .

It's important to use comments to explain your code and show that you're able to take ownership of the exercises and discuss them.

You are not expected to collaborate outside of the group on exercises and submitting other groups' code as your own will result in 0 points.

For questions contact: *riccardo.cusinato@unibe.ch* with the subject: *Programming for Data Science 2024*.

Deadline: 14:00, April 18, 2024.

Exercise - CO2 emissions

Re-create the plots found in 'plots.png' using the data found in 'owid-co2-data-reduced.csv' (source: <https://github.com/owid/co2-data>).

You do not have to make perfect replicas of the plots, but you should show that you are able to use the concepts shown in the lecture. If there are aspects of the plots you consider not to be reproducible given the information provided, comment this in your code. For instance, the size of dots may not be possible to reproduce exactly, and you will be awarded points for showing you are able to modify this parameter, where relevant, and pointing this out in your comments, rather than for reproducing the exact same dot size. The plots are made using matplotlib and seaborn, working on the dataframe.

Relevant concepts for exercise completion are:

- Saving
- Subplots (Object-oriented interface)
- Setting colors
- Adjusting axes limits
- Labels and legend
- Using seaborn together with pandas and matplotlib

Specific plot types requested:

- Line plot
- Violin plot
- Scatter plot

Useful tips:

- to select only the actual countries, check those that have an *iso_code* in the dataframe.
- the values "per capita" are obtained by normalizing to the population.
- the "percentage" values are obtained by normalizing on the total emissions.

NB: if you do not have seaborn installed, install it through *pip install seaborn*.